

I. Rappel.

L'écriture d'un nombre se fait en utilisant des symboles ou caractères. La base est le nombre de caractères différents dont on dispose pour son écriture.

Systèmes de numération Base	Symboles utilisés
Système de numération binaire Base 2	0 - 1
Système de numération décimal Base 10	0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9
Systèmes de numération hexadécimal Base 16	0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - A - B - C - D - E - F

En base 2, les symboles utilisés 0 et 1 sont appelés BIT (Binary digIT)

En base 10, les symboles utilisés sont les chiffres arabes.

En base 16, on utilise les chiffres arabes plus les 6 premières lettres de l'alphabet où :

$$\begin{array}{lll}
 A_{(H)} = 10_{(10)} & B_{(H)} = 11_{(10)} & C_{(H)} = 12_{(10)} \\
 D_{(H)} = __{(10)} & E_{(H)} = __{(10)} & F_{(H)} = __{(10)}
 \end{array}$$

II. Système de numération binaire.

II.1. Notion de mots

Un ensemble de bits est appelé mot. On rencontrera généralement :

- des mots de 8 bits (appelés encore octets). Exemple : 0010 1000
- des mots de 16 bits. Exemple : 0110 1100 0110 1010
- des mots de 32 bits. Exemple : 0010 1101 1110 1010 1111 0000 1001 1100

Lors de l'écriture d'un mot :

Le bit situé le plus à gauche est appelé **bit de poids fort ou MSB (Most Significant Bit)**

Le bit situé le plus à droite est appelé **bit de poids faible ou LSB (Less Significant Bit)**

Exemple : 1001 1010 1 est le _____ 0 est le _____

II.2. Conversion binaire > décimal

Pour convertir un nombre binaire en décimal, on additionne les chiffres binaires affectés de leurs poids respectifs (puissance de 2).

$$Ex : 10110_{(2)} = 0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 = 22_{(10)}$$

Autres exemples :

$$\begin{array}{l}
 110010_{(2)} = \underline{\hspace{2cm}} = __{(10)} \\
 1100101_{(2)} = \underline{\hspace{2cm}} = __{(10)}
 \end{array}$$

II.3. Conversion décimal > binaire

Pour convertir un nombre décimal en binaire, on écrit en partant de la droite les restes successifs de la division de ce nombre par 2.

Exemple :

$$\begin{array}{r}
 172 \quad | \quad 2 \\
 \hline
 \end{array}$$

$172_{(10)} = __{(2)}$

III. Système de numération hexadécimal

III.1. Introduction

Le système hexadécimal est utilisé car il permet une écriture simplifiée (plus compacte) des nombres binaires. Chaque groupe de 4 bits est remplacé par un chiffre hexadécimal selon le tableau de conversion ci-dessous :

Binaire	Hexadécimal	Décimal	Binaire	Hexadécimal	Décimal
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	A	10
0011	3	3	1011	B	11
0100	4	4	1100	C	12
0101	5	5	1101	D	13
0110	6	6	1110	E	14
0111	7	7	1111	F	15

III.2. Conversion binaire > hexadécimal

Pour convertir un nombre binaire en hexadécimal, on regroupe les bits par groupe de quatre en partant des poids faibles, puis on convertit chaque groupe de 4 bits en écrivant son symbole hexadécimal.

Ex : **1011 0101**₍₂₎ > **B 5**_(H)

Autres exemples :

1001 1101₍₂₎ > ______(H)

10 0101₍₂₎ > ______(H)

10010010111₍₂₎ = _____₍₂₎ > ______(H)

III.3. Conversion hexadécimal > binaire

Pour convertir un nombre hexadécimal en binaire, on convertit chaque symbole hexadécimal en binaire. La concaténation de l'ensemble converti donne le nombre binaire.

Ex : 1A_(H) = ?

1_(H) → 0001₍₂₎

A_(H) → 1010₍₂₎

D'où 1A_(H) = 0001 1010₍₂₎

Autres exemples :

C5_(H) = _____₍₂₎

8F_(H) = _____₍₂₎

III.4. Conversion hexadécimal > décimal

Pour convertir un nombre hexadécimal en décimal, on additionne les valeurs décimales des chiffres hexadécimaux multipliés par leurs poids respectifs (puissance de 16).

Ex : 1A_(H) = **1x16¹** + **10x16⁰** = 26₍₁₀₎

Autres exemples :

2B_(H) = _____ = _____₍₁₀₎

10D_(H) = _____ = _____₍₁₀₎

III.5. Conversion décimal > hexadécimal

Pour convertir un nombre décimal en hexadécimal, on écrit en partant de la droite les restes successifs de la division de ce nombre par 16.

Exemple : 468₍₁₀₎ = _____₍₁₆₎

IV. Le code ASCII (American Standard Code for Information Interchange)

ASCII ⇒ American Standard Code for Information Interchange. La table ASCII est une norme d'encodage informatique des caractères alphanumériques de l'alphabet latin. La norme ASCII (on prononce phonétiquement "aski") établit une correspondance entre une représentation binaire des caractères de l'alphabet latin et les symboles, les signes, qui constituent cet alphabet. Par exemple, le caractère "a" est associé à "01100001" et "A" à "01000001".

La norme ASCII permet ainsi à toutes sortes de machines de stocker, analyser et communiquer de l'information textuelle.

L'ASCII étendu a permis à moindre coût de déployer mondialement des logiciels représentant le texte en peu d'octets et d'ignorer tout ou partie des problématiques d'internationalisation. Il a également conduit à des problèmes d'interopérabilité, qui ont abouti à l'émergence de standards tels qu'Unicode.

La table ASCII est internationale jusqu'au caractère 128, ensuite il diffère selon les pays.

IV.1. Les caractères de contrôle

Les caractères de contrôle sur la table ASCII se trouvent de 0 à 31. Ils ont chacun une signification particulière, ce langage est un langage de programmation, il sert donc à communiquer avec l'ordinateur.

Sur l'image ci-contre, est répertoriée la description des 31 premiers caractères. On peut voir par exemple le (00000001) qui signifie début de l'entête, puis plus loin (00000011) qui signifie la fin de la transmission de données. Il peut aussi avoir des données sonores comme (00000111) qui est une sonnerie (comme les bips que l'on peut parfois entendre sur nos ordinateurs).

Décimal	Hex	Binaire	Caractère
000	00	00000000	NUL (Null char.)
001	01	00000001	SOH (Start of Header)
002	02	00000010	STX (Start of Text)
003	03	00000011	ETX (End of Text)
004	04	00000100	EOT (End of Transmission)
005	05	00000101	ENQ (Enquiry)
006	06	00000110	ACK (Acknowledgment)
007	07	00000111	BEL (Bell)
008	08	00001000	BS (Backspace)
009	09	00001001	HT (Horizontal Tab)
010	0A	00001010	LF (Line Feed)
011	0B	00001011	VT (Vertical Tab)
012	0C	00001100	FF (Form Feed)
013	0D	00001101	CR (Carriage Return)
014	0E	00001110	SO (Shift Out)
015	0F	00001111	SI (Shift In)
016	10	00010000	DLE (Data Link Escape)
017	11	00010001	DC1 (XON)(Device Control 1)
018	12	00010010	DC2 (Device Control 2)
019	13	00010011	DC3 (XOFF)(Device Control 3)
020	14	00010100	DC4 (Device Control 4)
021	15	00010101	NAK (Negative Acknowledgement)
022	16	00010110	SYN (Synchronous Idle)
023	17	00010111	ETB (End of Trans. Block)
024	18	00011000	CAN (Cancel)
025	19	00011001	EM (End of Medium)
026	1A	00011010	SUB (Substitute)
027	1B	00011011	ESC (Escape)
028	1C	00011100	FS (File Separator)
029	1D	00011101	GS (Group Separator)
030	1E	00011110	RS (Request to Send)(Record Separator)
031	1F	00011111	US (Unit Separator)
032	20	00100000	SP (Space)
033	21	00100001	! (exclamation mark)
034	22	00100010	" (double quote)
035	23	00100011	# (number sign)
036	24	00100100	\$ (dollar sign)
037	25	00100101	% (percent)
038	26	00100110	& (ampersand)
039	27	00100111	' (single quote)
040	28	00101000	((left opening parenthesis)
041	29	00101001) (right closing parenthesis)
042	2A	00101010	* (asterisk)
043	2B	00101011	+ (plus)
044	2C	00101100	, (comma)
045	2D	00101101	- (minus or dash)
046	2E	00101110	. (dot)
047	2F	00101111	/ (forward slash)
048	30	00110000	0
049	31	00110001	1
050	32	00110010	2
051	33	00110011	3
052	34	00110100	4
053	35	00110101	5
054	36	00110110	6
055	37	00110111	7
056	38	00111000	8
057	39	00111001	9
058	3A	00111010	: (colon)
059	3B	00111011	; (semi-colon)
060	3C	00111100	< (less than sign)
061	3D	00111101	= (equal sign)
062	3E	00111110	> (greater than sign)
063	3F	00111111	? (question mark)

IV.2. Les caractères spéciaux et nombres

Les caractères de contrôle sur la table ASCII se trouvent de 32 à 63. Cette partie de la table sert à pouvoir insérer des caractères spéciaux, comme des points d'exclamations (!) et (00100001) en binaire ou un égal (=) et (00111101) en binaire. On peut utiliser le langage ASCII, il suffit de presser alt puis de composer le numéro décimal qui se trouve après le &#.

IV.3. Les lettres capitales

Cette section du tableau permet de mettre du texte en majuscule, comme U(U)(01010101) ou encore E(E)(01000101). Il y a aussi quelques caractères spéciaux que l'on peut utiliser avec des majuscules, comme @(@)(01000000).

064	40	01000000	@	(AT symbol)
065	41	01000001	A	
066	42	01000010	B	
067	43	01000011	C	
068	44	01000100	D	
069	45	01000101	E	
070	46	01000110	F	
071	47	01000111	G	
072	48	01001000	H	
073	49	01001001	I	
074	4A	01001010	J	
075	4B	01001011	K	
076	4C	01001100	L	
077	4D	01001101	M	
078	4E	01001110	N	
079	4F	01001111	O	
080	50	01010000	P	
081	51	01010001	Q	
082	52	01010010	R	
083	53	01010011	S	
084	54	01010100	T	
085	55	01010101	U	
086	56	01010110	V	
087	57	01010111	W	
088	58	01011000	X	
089	59	01011001	Y	
090	5A	01011010	Z	
091	5B	01011011	[(left opening bracket)
092	5C	01011100	\	(back slash)
093	5D	01011101]	(right closing bracket)
094	5E	01011110	^	(caret circumflex)
095	5F	01011111	_	(underscore)
096	60	01100000		
097	61	01100001	a	
098	62	01100010	b	
099	63	01100011	c	
100	64	01100100	d	
101	65	01100101	e	
102	66	01100110	f	
103	67	01100111	g	
104	68	01101000	h	
105	69	01101001	i	
106	6A	01101010	j	
107	6B	01101011	k	
108	6C	01101100	l	
109	6D	01101101	m	
110	6E	01101110	n	
111	6F	01101111	o	
112	70	01110000	p	
113	71	01110001	q	
114	72	01110010	r	
115	73	01110011	s	
116	74	01110100	t	
117	75	01110101	u	
118	76	01110110	v	
119	77	01110111	w	
120	78	01111000	x	
121	79	01111001	y	
122	7A	01111010	z	
123	7B	01111011	{	(left opening brace)
124	7C	01111100		(vertical bar)
125	7D	01111101	}	(right closing brace)
126	7E	01111110	~	(tilde)
127	7F	01111111	DEL	(delete)

IV.4. Les lettres minuscules

Cette section est la même que les lettres en capitales sauf qu'ici elles seront en minuscule, cela permet d'écrire un texte. On commence avec la commande (00000010) puis on écrit notre texte avec cette section de tableau, comme par exemple d(d) (01100100), puis on finit avec la commande (00000011).

table ascii étendue (128 - 255)

IV.5. Table ASCII étendue

La table ASCII étendue dépend de chaque pays, celle-ci est la française, elle permet d'avoir d'autre caractères du type §(õ) (11110101) ou encore ±(ñ) (11110001). Il existe le caractère 255 qui n'est pas considéré comme un espace mais comme un espace vide, on peut l'utiliser par exemple dans les mots de passe où il est spécifié que l'on ne peut utiliser d'espace.

Exemple : envoi du caractère A

Au caractère A correspond 41_(H)
Ainsi quand on appuie sur A,
on envoie l'octet 0100 0001

Autres exemples :

Envoi du caractère ?

Au caractère ? correspond _____(H)
Ainsi quand on appuie sur ?,
on envoie l'octet _____

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	ā	192	C0	Ł	224	E0	α
129	81	Û	161	A1	ī	193	C1	ł	225	E1	β
130	82	ü	162	A2	ı	194	C2	Ł	226	E2	γ
131	83	ē	163	A3	Ű	195	C3	ł	227	E3	δ
132	84	ā	164	A4	ñ	196	C4	—	228	E4	ε
133	85	ā	165	A5	ñ	197	C5	—	229	E5	σ
134	86	ā	166	A6	ñ	198	C6	—	230	E6	μ
135	87	Ĉ	167	A7	ε	199	C7	—	231	E7	γ
136	88	ĉ	168	A8	ı	200	C8	—	232	E8	φ
137	89	ĕ	169	A9	ı	201	C9	—	233	E9	β
138	8A	ĕ	170	AA	½	202	CA	—	234	EA	Ω
139	8B	ĭ	171	AB	½	203	CB	—	235	EB	δ
140	8C	ĭ	172	AC	¼	204	CC	—	236	EC	≈
141	8D	—	173	AD	ı	205	CD	—	237	ED	φ
142	8E	—	174	AE	ı	206	CE	—	238	EE	ε
143	8F	Ā	175	AF	»	207	CF	—	239	EF	∩
144	90	Ē	176	B0	—	208	D0	—	240	F0	≡
145	91	æ	177	B1	—	209	D1	—	241	F1	±
146	92	ŕ	178	B2	—	210	D2	—	242	F2	≥
147	93	ō	179	B3	—	211	D3	—	243	F3	≤
148	94	ö	180	B4	—	212	D4	—	244	F4	∫
149	95	ö	181	B5	—	213	D5	—	245	F5	∫
150	96	ü	182	B6	—	214	D6	—	246	F6	÷
151	97	Û	183	B7	—	215	D7	—	247	F7	≈
152	98	Û	184	B8	—	216	D8	—	248	F8	°
153	99	Û	185	B9	—	217	D9	—	249	F9	•
154	9A	Û	186	BA	—	218	DA	—	250	FA	·
155	9B	Û	187	BB	—	219	DB	—	251	FB	√
156	9C	Û	188	BC	—	220	DC	—	252	FC	•
157	9D	Û	189	BD	—	221	DD	—	253	FD	•
158	9E	Û	190	BE	—	222	DE	—	254	FE	•
159	9F	f	191	BF	—	223	DF	—	255	FF	•

Exercice de passage entre codes décimal, binaire naturel, hexadécimal et BCD

(Tableau à compléter)

Décimal	Binaire naturel	Hexadécimal
0	0000	0
1		
2		
3		
4		
5		
6		
7		
8		
9		
10	1010	A
11		
12		
13		
14		
15		
16	1 0000	10
		12
	10 0000	
		64
127	1000 0000	
		FF
	1 0101 1110	
512		3E8
	100 0000 0000	
		93D
	1 1100 1101 1110	
9570		